CS 3240A / CS 5240A Fall 2025
Topic review for the midterm exam

You may prepare and use a single 8.5" x 11" sheet (both sides) with notes and formulas; other than this, no notes or books.

* Graphs: definitions, characteristics
  - paths and cycles
  - directed vs. undirected
  - connectedness; connected components (of an undirected graph)
  - strongly connected components (of a directed graph)
  - BFS and DFS traversal; tracing the algorithms
  - bipartite graphs: definition; determining bipartiteness
  - topological ordering of directed graphs: definition, implications, algorithm

* Greedy algorithms
  - Interval scheduling: be able to trace the algorithm
  - Dijkstra's Algorithm: be able to trace the algorithm
  - minimum spanning trees: characteristics; tracing Prim's and Kruskal's algorithms
  - be able to describe and apply (but not prove) the cycle property and the cut property
  - graduate students: be able to understand why Dijkstra's algorithm produces correct results
  - graduate students: be able to explain why the cycle property and the cut property hold

* Divide and tackle
  - recurrence relations: understanding a recurrence relation; being able to write a recurrence relation given the description of a divide-and-tackle algorithm
  - unrolling trees: drawing an unrolling tree; showing the total work at each level
  - steps involved in showing a bound for T(n) using an unrolling tree
  - graduate students: be able to set up and complete a proof by induction for a simple recurrence relation

Example questions

* true or false: a bipartite graph cannot have a cycle; if true, then explain why; if not, then show a counterexample

* the *longest path* between nodes u and v in a graph is the longest simple path (no vertex visited more than once) between u and v. In a connected undirected graph G, what structure must the G have if the longest path between any two nodes has the

same length as the shortest path between those two nodes?

* given the following undirected, weighted graph [and there would be a graph], trace Dijkstra's algorithm starting at node a [and there would be a table of the form that we used in the lecture]

* given the following graph [and there would be a graph], show the order of edges selected by Prim's algorithm